

OHRI Package on 3.x RefApp

- Ticket for the first pass at this work:
<https://issues.openmrs.org/browse/O3-1134>

Situation:

- **3.x RefApp:** We have a dev & demo environment for the 3.x RefApp (dev3.openmrs.org & o3.openmrs.org)
- **OHRI Package:** We have a demo environment for the OHRI assets (ohri-demo.globalhealthapp.net/openmrs/spa) (this will still be necessary so that OHRI has it's own place, a single place to show "here's the OHRI-specific work")
- **3.x RefApp + OHRI Package:** We **don't** have a place where the RefApp and OHRI package are stuck together to see what happens (like an example of a distribution adding the OHRI package).
 - Sandbox GitHub Actions setup:
<https://github.com/UCSF-IGHS/openmrs-esm-ohri/blob/master/.github/workflows/node.js.yml>

For the OHRI "package", 4 things are already available:

1. **Frontend OHRI App**
 - a. (grab "latest" version from <https://www.npmjs.com/package/openmrs-ohri-app>)
We can tie the environment to this OHRI App, so updates to the npm package are auto-updated in
2. **OHRI Backend module**
 - a. <https://github.com/UCSF-IGHS/openmrs-module-ohri-core>
3. **OHRI Concept collections on OCL**
 - a. We can import these from here:
<https://app.openconceptlab.org/#/orgs/OHRITEchGroup/collections/>
4. **OHRI Iniz file** (all other non-concept metadata)
 - a. <https://github.com/UCSF-IGHS/openmrs-distro-ohri/tree/OHRI-49-Iniz-package-for-HTS/package/config/configuration>

Proposal: **Let's take the 4 existing OHRI artifacts and try putting these onto a server with the 3.x RefApp (e.g. "[ohri.o3.openmrs.org](https://issues.openmrs.org/browse/O3-1134)").** Set up in a way that it auto-updates whenever the OHRI package is updated. The process is an experiment and subject to change based on collaborative feedback about the experience.

This would help us achieve these goals:

1. We can show the world that the 2 work together, and

2. We can catch any rough edges where the 2 conflict in any way (e.g. if something is pushed to the RefApp that would break/impact the OHRI package, and vice-versa), and
3. We have a first proof-of-concept example of a “package” :)

What's Needed:

- **Existing assets from UCSF:** (Done)
 - Share Iniz config file
 - Way to download Backend Module (needs to be published to maven repo somewhere)
- **OMRS:**
 - Set up new environment, e.g. ohri.o3.openmrs.org, on OMRS infrastructure, using 3.x RefApp
 - Add openmrs-ohri-app to distro.properties for that environment
 - Give access to Sr UCSF dev(s) to Bamboo build so this can be updated whenever
 - Set up GHA so openmrs-ohri-